NPS-AM-09-042

EXCERPT FROM THE PROCEEDINGS

OF THE

SIXTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

APPLICATION OF REAL OPTIONS THEORY TO SOFTWARE-INTENSIVE SYSTEM ACQUISITIONS

Published: 22 April 2009

by

Capt Albert Olagbemiro, Man-Tak Shing and Johnathan Mun

6th Annual Acquisition Research Symposium of the Naval Postgraduate School:

Volume II: Defense Acquisition in Transition

May 13-14, 2009

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **APR 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**
4. TITLE AND SUBTITLE **Application of Real Options Theory to Software-intensive System Acquisitions**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Computer Science,Monterey,CA,93943**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

In the Department of Defense (DoD), the typical outcome of a software acquisition program has been massive cost-escalation, slipping planned delivery dates and making major cuts in the planned software functionality to guarantee program success. To counter this dilemma, the DoD put forth a new weapons acquisition policy in 2003 based on an evolutionary acquisition approach to foster increased efficiency while building flexibility in the acquisition process. However, the evolutionary acquisition approach often relies on the spiral development process, which assumes the end-state requirements are known at the inception of the development process, a misrepresentation of reality in the acquisition of DoD software-intensive weapons systems. This article presents a framework to address the issue of requirements uncertainty as it relates to software acquisition. The framework is based on Real Options theory and aims at mitigating risks associated requirement volatility based on the technology objectives, constraints as put forth by the customer at the acquisition decision-making level.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**62**	

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

Proceedings of the Annual Acquisition Research Program

The following article is taken as an excerpt from the proceedings of the annual Acquisition Research Program. This annual event showcases the research projects funded through the Acquisition Research Program at the Graduate School of Business and Public Policy at the Naval Postgraduate School. Featuring keynote speakers, plenary panels, multiple panel sessions, a student research poster show and social events, the Annual Acquisition Research Symposium offers a candid environment where high-ranking Department of Defense (DoD) officials, industry officials, accomplished faculty and military students are encouraged to collaborate on finding applicable solutions to the challenges facing acquisition policies and processes within the DoD today. By jointly and publicly questioning the norms of industry and academia, the resulting research benefits from myriad perspectives and collaborations which can identify better solutions and practices in acquisition, contract, financial, logistics and program management.

For further information regarding the Acquisition Research Program, electronic copies of additional research, or to learn more about becoming a sponsor, please visit our program website at:

www.acquistionresearch.org

For further information on or to register for the next Acquisition Research Symposium during the third week of May, please visit our conference website at:

www.researchsymposium.org

THIS PAGE INTENTIONALLY LEFT BLANK

Application of Real Options Theory to Software-intensive System Acquisitions

Presenter: Capt Albert Olagbemiro, PhD, is a Logistics Readiness Officer in the US Air Force Reserve and has served in both staff and operational logistics assignments. He is currently the Operations Officer at the 69th Aerial Port Squadron, Andrews AFB, MD. He holds both an MS in Computer Science and an MBA from Johns Hopkins University. He received his PhD in Software Engineering from the Naval Postgraduate School in 2008.

Capt Albert Olagbemiro USAFR
Computer Science Department, Naval Postgraduate School, Monterey, CA 93943
443-803-9788, albert.olagbemiro@afncr.af.mil

Authors:

Dr. Man-Tak Shing is an associate professor at the Naval Postgraduate School. His research interests include the engineering of software intensive systems. He is on the program committees of several software engineering conferences and is a member of the Steering Committee of the IEEE International Rapid System Prototyping Symposium. He received his PhD in computer science from the University of California, San Diego, and is a senior member of IEEE.

Man-Tak Shing
Computer Science Department, Naval Postgraduate School, Monterey, CA 93943
831-656-2634, shing@nps.edu

Dr. Johnathan Mun is a research professor at the Naval Postgraduate School and Founder/CEO of Real Options Valuation, Inc. (software, training and consulting firm in Silicon Valley, focusing on advanced risk analytics, risk simulation, stochastic forecasting, portfolio optimization, strategic real options analysis, and general business modeling analytics). He received his PhD and MBA from Lehigh University and holds other financial charters and certifications in risk and financial analysis.

Johnathan Mun
Information Sciences Department, Naval Postgraduate School, Monterey, CA 93943
925-271-4438, jcmun@realoptionsvaluation.com

Abstract[1]

In the Department of Defense (DoD), the typical outcome of a software acquisition program has been massive cost-escalation, slipping planned delivery dates and making major cuts in the planned software functionality to guarantee program success. To counter this dilemma, the DoD put forth a new weapons acquisition policy in 2003 based on an evolutionary acquisition approach to foster increased efficiency while building flexibility in the acquisition process. However, the evolutionary acquisition approach often relies on the spiral development process, which assumes the end-state requirements are known at the inception of the development process, a misrepresentation of reality in the acquisition of DoD software-intensive

[1] This work was supported in part by the NPS Acquisition Research Program—OUSD_08 (Project #:F08-023, JON: RGB58). The views and conclusions in this talk are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the US Government.

weapons systems. This article presents a framework to address the issue of requirements uncertainty as it relates to software acquisition. The framework is based on Real Options theory and aims at mitigating risks associated requirement volatility based on the technology objectives, constraints as put forth by the customer at the acquisition decision-making level.

1. Introduction

The software acquisition lifecycle, which encapsulates the activities related to its procurement, development, implementation and subsequent maintenance, continues to present challenges to software executives and program managers due to increasingly complex organizational requirements and the ever increasing role which software plays in U.S. Department of Defense (DoD) weapons systems. Various factors and considerations, most of which are complex in nature compound the software acquisition process, factors which present themselves in the form of "uncertainties", and which have the potential of introducing risks if the uncertainties are not adequately addressed and or resolved. In this paper, we address the issue of requirements uncertainty and propose a methodology for addressing this issue. Our approach addresses these issues by taking a proactive/preemptive approach to risk management by planning and paying for the risks associated with requirements up front. This is not to say that risk management strategies are not being adopted today, but rather a failure of management to take a strategic approach towards risk management. The status quo today is to employ reactive risk management strategies that often result in the reduction of much needed functionality from the scope of the software investment effort. We therefore propose a more proactive decision-making framework that involves identifying the risks, pricing risk upfront during the planning stages of the acquisition before a decision to commit resources is made.

2. The Requirements Dilemma

In software development, requirements instability have been found to have a profound impact on a program's schedule and drive up costs due to increases in Research, Development, Test & Evaluation (RDT&E) costs associated with the requirements changes. The lack of adequately defined requirements is one of the leading problems in the software development effort. Without adequate definition and validation of requirements and design, software engineers could be coding to an incorrect solution, resulting in missing functionality and errors. This dilemma is highlighted in a 2007 interview of the Army's Program Executive Officer (PEO) for Ammunition, in which "the ability to acquire and maintain, safe, reliable supportable and modifiable software systems which met user requirements in an environment of rapid technological advances" was identified as their biggest challenge in software acquisitions (Starrett 2007)[2]. Furthermore, the U.S. Government Accountability Office (GAO) responsible for reviewing weapon systems investments found consistent problems of cost increases, schedule delays, and performance shortfalls exacerbated by factors such as pressure on program managers to promise more than they could deliver. These concerns infer a resounding theme which continues to be resonated within the software acquisition community: *Meeting customer requirements within cost and schedule constraints*.

---

[2] Starrett, E. (2007). Software Acquisition in the Army. *Crosstalk: The Journal of Defense Software Engineering*, 20(5), 4-8.

Balancing the satisfaction of a customer's ever-changing requirements within the realms of meeting both current and future uncertain operational needs against the costs and schedule constraints poses a cumbersome challenge to the software executive, thereby making software-investments a very risky venture; risky in the sense that software engineering and investment decisions are plagued by uncertainties which more often than not leads to varying degrees of risk ranging from operational shortfalls to cost and schedule overruns.

Ever changing requirements continues to impact software acquisition efforts, and more often than not, forces managers to choose between requirements, i.e., which requirements to accept and which requirements to reject with the full understanding that ignoring changes in requirements has the consequence of the delivered product failing to meet the customers needs while accepting changes in requirements has the potential of impacting costs and schedule.

Furthermore, changes in requirements while a software acquisition effort is underway also poses the risk of introducing unwanted, unanticipated or unknown impact on existing requirements, not to mention the associated costs and scheduled delays depending on the phase of the investment or software development process experiencing significant requirements changes. While the standard practice has been to "freeze" requirements prior to the commencement of any development activities, more often than not, this does not work and is also not representational of the DoD doctrine to support the flexible development and rapid delivery of products to meet the war-fighters needs in an ever changing environment in response to operational needs.

The inefficiencies of current management techniques as shown in Table 1 highlight the needs of new management approaches that proactively plan for, and factor in uncertainty into their acquisition strategy. This is because the acquisition of software, its development and the operational use of the software are all dominated by human action, human judgment and decision making, and inevitably human error. The outcome is, therefore, often uncertain and unpredictable, and leads to unavoidable uncertainties which introduces and drives risk (Starrett, 2007).

Program	**Initial Investment**	**Initial Quantity**	**Latest Investment**	**Latest Quantity**	**% Unit Cost Increase**	**% Quantity Decrease**
Joint Strike Fighter	$189.8 billion	2,866 aircraft	$206.3 billion	2,459 aircraft	26.7	14.2
Future Combat Systems	$92 billion	18 System	$163.7 billion	14 systems	54.4	22.3
F-22A Raptor	$81.1 billion	648 aircraft	$65.4 billion	181 aircraft	188.7	72.1

Table 1. Program Management Failures of Top Three Major Weapons Systems. [3]

We must however emphasize that uncertainty should not be confused with risk as there is an important distinction between the two. Risk is something one bears and is the outcome of uncertainty, as uncertainty is either resolved through the passage of action or left unattended due to inaction (Mun, 2006)[4]. The risks associated with the acquisition of the software need to

[3] Numbers were complied from various GAO reports and were current as of 2007.

[4] Mun, J. (2006). *Real Options Analysis (Second Edition)* Wiley.

be identified and analyzed very early on in the decision-making process, and an approach to mitigate the high-priority risks must be incorporated into a software acquisition plan.

Therefore in order to accurately estimate requirements volatility and its impact on the future value of a software-intensive-system under consideration for acquisition, not only must the risk of requirements changes be quantified, it must also be specifically predicted and quantified based on the phase in the software development process in which the changes are more likely to occur. Hence the need for an approach that would explicitly acknowledge not only the probability of occurrence based on previous objective estimates, but also the possibility of occurrence based on subject expert opinions (Delphi Method) that acknowledges either the degree of belief or ignorance in the objective probability estimates. (See Section 4 for details.)

3. The Real Options Approach

The Real Options approach is based on the concepts of financial options theory, and it builds on several tried-and-proven approaches of management. In the study conducted in (Olagbemiro 2008)[5], it was shown how it could be used as proactive risk management tool within a strategic decision-making level (executive level) pre-acquisition context, further complementing the spiral development approach at the “tactical level”. It was also demonstrated using the U.S. Army Future Combat Systems program as an example of how the traditional Real Options methodology, when enhanced and properly formulated around a proposed or existing software-investment, could provide a framework for guiding software acquisition decision-making by highlighting the strategic importance of managerial flexibility. This flexibility offers management the ability to balance the satisfaction of a customer’s requirements within the realms of the associated cost and schedule constraints by developing the appropriate options during the acquisition decision making phase and executing the options when it becomes optimal to do so. However, the Real Options approach calls for the existence or satisfaction of certain pre-conditions before it can be applied. These pre-conditions, which correlate directly to the various activities associated with software related capital investments, are outlined in (Mun 2006) as follows:

1. The existence of a basic financial model used to evaluate the costs and benefits of the underlying software asset (e.g. Net Present Value (NPV) as the Real Options approach builds on the existing tried-and-tested approaches of current financial modeling techniques.

2. The existence of uncertainties during the software-related capital investment decision-making process otherwise, the Real Options analysis becomes useless as everything is assumed to be certain and known.

3. The uncertainties surrounding the software-related capital investment decision-making process must introduce risks which directly impact the decision-making process. Real Options could then be used to hedge the downside risk and take advantage of the upside uncertainties.

---

[5] Olagbemiro, A. (2008). *Application of Real Options Theory to Software Engineering for Strategic Decision Making in Software Related Capital Investments*. PhD Dissertation, Naval Postgraduate School, Monterey, California.

4. Management must have the flexibility or option to make mid course corrections when actively managing the project.

5. Management must be smart enough to execute the Real Options when it becomes optimal to do so.

3.1 Real Options Valuation

Real options valuation originated from research performed to price financial option contracts in the field of financial derivatives. The underlying premise of its suitability and applicability to software engineering is based on the recognition that strategic flexibility in software acquisitions decisions can be valued as a portfolio of options or choices in real "assets", much akin to options on financial securities which have real economic value under uncertainty (Dixit & Pindyck, 1995)[6]. In contrast to financial options, real options valuation centers on real or non-financial assets, and is valuable because it enables the option holder (software program manager) to take advantage of potential upside benefits while controlling and hedging risks. When extended to a real "asset" such as software, real options could be used as a decision-making tool in a dynamic and uncertain environment. An option gives its holder the *rights but without the obligations*, to acquire or dispose of a risky asset at a set *price* within a specified time period (Erdogmus, 1999)[7]. If the market conditions are favorable before the option expires, the holder exercises this right, thus making a profit, otherwise, the holder lets the option expire.

A necessary and key tenet of the real options approach is a requirement for the presence of uncertainties, a constraint that is widely characteristic of software acquisitions decision-making. Software acquisitions encapsulate the activities related to software procurement, development, implementation, and subsequent maintenance. The uncertainties which surround these activities are compounded by increasingly complex requirements demanded by the warfighter and present themselves in various forms ranging from changing or incomplete requirements, insufficient knowledge of the problem domain, decisions related to the future growth, technology maturation and evolution of the software.

To tackle the issue, we developed a formal and distinct uncertainty elicitation task as part of the software investment decision-making process (Figure 1) to obtain information on the relevant uncertainties from a strategic point of view. While this task would not include members of a typical requirements team, they would work in tandem with the requirements team, to identify and document uncertainties as they are revealed from an independent point of view. Implementing an explicit uncertainty elicitation task would facilitate the identification of uncertainties very early on in the acquisition process, so that the necessary steps could be taken to either refine the requirements to address the uncertainties or identify strategic options to mitigate the risks posed by the uncertainties.

---

[6] Dixit, A.K. and Pindyck, R.S. (1995). The options approach to capital investment. *Harvard Business Review,* 73(3), 105-115.

[7] Erdogmus, H. (1999). Valuation of Complex Options in Software Development, *Proc. ICSE'99 Workshop on Economics Driven Software Engineering Research (EDSER1)*, Los Angeles, CA.

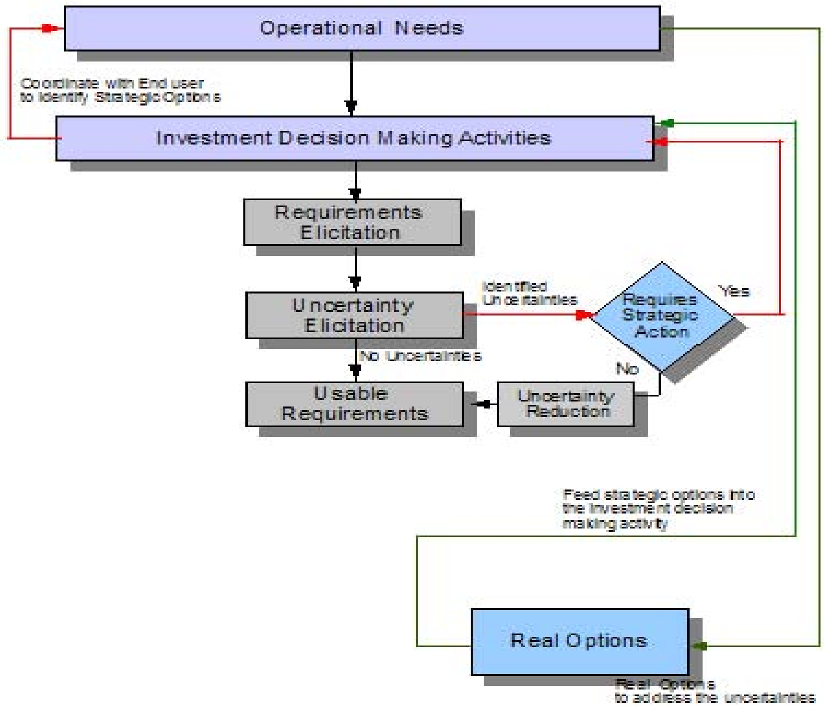


Figure 1. Uncertainty Elicitation Model

In the uncertainty elicitation step in the model, uncertainties are captured from two perspectives (the managerial and technical perspective) using what we call the "2 **T**" approach as illustrated in Figure 2. Managerial uncertainties of people, time, functionality, budget, and resources contribute to both estimation and schedule uncertainties which are considered to be pragmatic uncertainties. Technical uncertainties of incomplete requirements, ambitious, ambiguous, changing or unstable requirements contribute to software specification uncertainties, which lead to software design and implementation, software validation and software evolution uncertainties all of which can be categorized as exhibiting both Heisenberg-type and Gödel-like uncertainties.

If uncertainty cannot be resolved, strategic real options could be developed to address the risks posed by the uncertainty, providing management the flexibility to address the risks posed by the uncertainties when they become revealed at a later date during the acquisition effort.

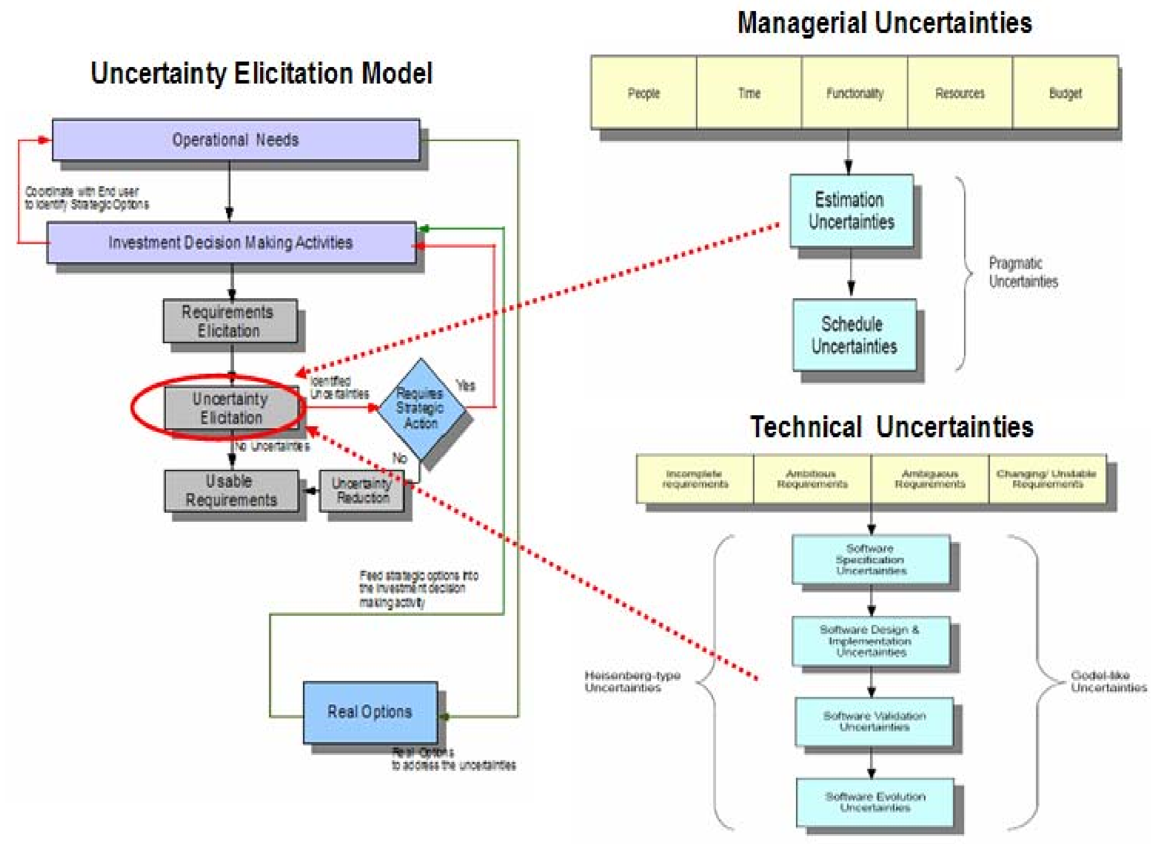


Figure 2. Expanded View of Uncertainty Elicitation Model

3.2 The Real Options Framework

To develop the appropriate options to hedge against the risks due to the uncertainties surrounding a software acquisition effort, we develop a generalized Real Options Framework (Figure 3) in line with the 5 preconditions as outlined in (Mun, 2006). This proposed framework consists of the following four phases each of which explicitly addresses and establishes compliance with the preconditions.

1. Study Phase
2. Data Collection and Preparation Phase
3. Analysis Phase
4. Execution Phase

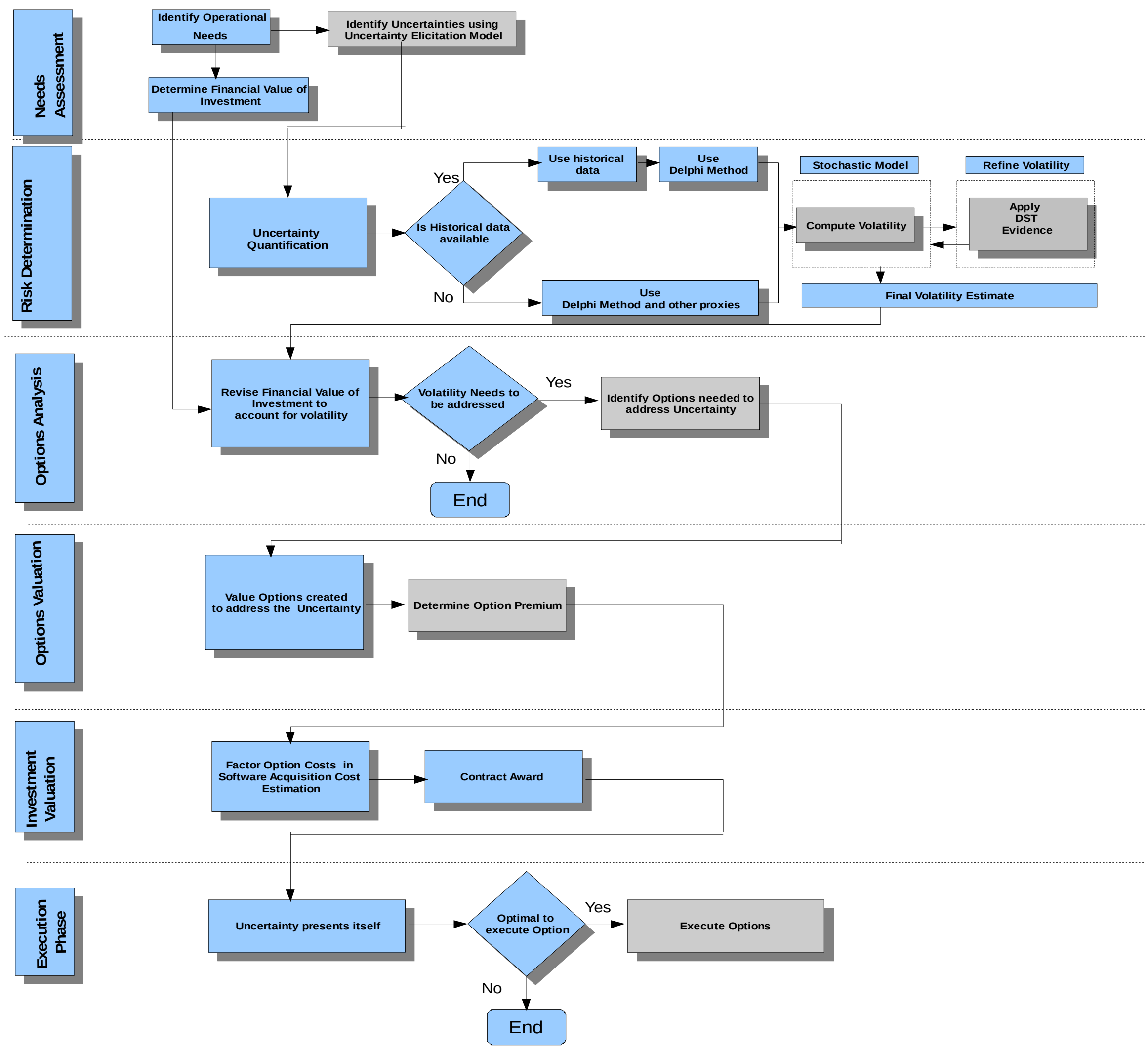


Figure 3. Real Options Framework

4. Addressing Uncertainty

Uncertainties permeate virtually every phase of the software acquisition process ranging from procurement decision-making, requirements specification, software development and implementation, to the eventual evolution of the software. These uncertainties could be broadly categorized into the categories shown in Figure 4.

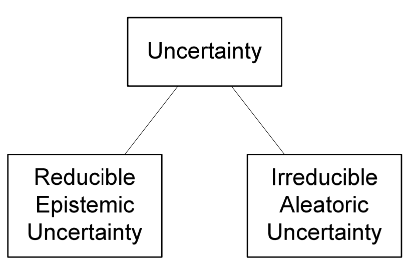


Figure 4. Taxonomy of Uncertainty

Epistemic uncertainties are reducible and they deal with our lack of knowledge, lack of information and our own and others' subjectivity concerning an issue. Aleatoric uncertainties, on the other hand, are irreducible and they deal with the randomness (or predictability) of an event due to variability of input or model parameters when the characterization of the variability is available (Wojtkiewicz et al., 2001)[8]. In other words, an aleatoric uncertainty is an inherent variation associated with the physical system or the environment. Both epistemic and aleotoric uncertainties are interwoven and form the general framework of uncertainties which plague software acquisition efforts from a requirements uncertainty perspective.

Since requirements uncertainty implies risk, consequently, uncertainty must be duly quantified as a risk factor to gauge the magnitude of its impact on the underlying asset. The process of translating or equating software engineering uncertainties into a quantifiable property begins with the quantification of the identified requirements uncertainties, computing the impact of uncertainties and ultimately developing a risk analysis framework in which the associated risks are identified, predicted and modeled using simulation and the results analyzed and costs factored into the software acquisition as appropriate.

4.1 Estimating Requirements Volatility

While volatility is just one of the parameters needed for Real Options analysis, it is the most difficult of all the parameters to estimate. Given the impact that requirements instability has on costs, we attempt to determine the rate of change of requirements or the volatility of requirements, and use volatility to quantify the risk of requirements changes in the proposed software acquisition effort.

In order to estimate the volatility of the returns associated with our current software investment effort, we attempt to gather evidence to help derive our estimates. Historically, gathering of evidence using previously completed software-related capital investments as a proxy is a difficult task for the following reasons:

1. The current software investment effort under consideration might be the first of its kind with no known comparables.
2. Information is rarely or actively collected and managed in a disciplined fashion.
3. Even when information is collected, accessibility by third parties is usually difficult due to the proprietary nature of the information.

Thus more often than not the software executive is faced with identifying alternate sources of information to either assert or dispute their initial volatility estimates. In our study, we propose to use either historical data (objective approach) or expert opinions obtained using the Delphi method (subjective approach). We choose to use both methods because we believe intuition and judgment (subjective approach) should supplement quantitative analysis (objective approach). More often than not, past success and failures serve as key indicators of the future.

---

[8] Wojtkiewicz, S.F., Eldred, M.S., Field, R.V., Urbina, A., and Red-Horse, J.R. (2001) Uncertainty Quantification In Large Computational Engineering Models. *Proc. 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Number AIAA-2001-1455.

Thus historical data can be used to predict and explore “what-if” scenarios on future projects based on the use of forecasting and analytical analysis.

The Delphi Method is a technique first introduced by the RAND Corporation in the 1940’s as a methodology for the elicitation of the opinion of an expert or groups of experts to guide decision making by the making predictions about future events. It places emphasis on an iterative, systematic, disciplined and interactive process of individual interviews (usually conducted using questionnaires) and the outcome is based on the Hegelian Principle of achieving consensus through a three step process of thesis, antithesis, and synthesis (Stuter, 1996)[9]. In the thesis and antithesis steps, the team of experts present their opinion or views on the given subject, establishing views and opposing views, and consensus is ultimately reached during the synthesis phase as opposing views are brought together to form the new thesis. Widely used as an estimating tool, the Delphi Method has been used to estimate values for factors which appear in software estimation models such as cost estimation (Boehm et al., 2000)[10] and risk estimation. Furthermore, it is one of the approved techniques published by the U.S. Army Cost and Economic Analysis Center in February 2001 for preparing or reviewing economic analyses in support of the decision making process.

In the event that there is no historical data available, the customer should resort to obtaining the required information using the Delphi Method. In the case that we do have historical data but are unable to find projects meeting any or all of the criteria above, we proceed to “fit” the data to as close as possible to mimic our current software investment effort by employing interpolation techniques to understand and forecast our project based on the trends depicted in the historical data.

To determine the rate of volatility, we employ the Caper Jones’ approach which is a transposition from the financial industry (Kulk & Verhoef, 2008)[11]. Jones asserts that existing methods of average percentage of change of the overall requirements volume lacks information, because it does not give any information on the time in which the change occurred, a key factor that is important to determine in software engineering, since requirements changes become more expensive to implement, the farther we are into the software development process.

Jones therefore uses the compound monthly requirements volatility rate to express the time aspect. Calculating monthly requirements volatility rates, as defined by Jones, is a transposition from the financial world. The time value or future value of money is well-known in the field of accounting as compound interest or CAGR, short for compound annual growth rate. By transposing from compound growth rate in finance we assume that requirements are compounded within a project (Kulk & Verhoef, 2008). The basic financial equation is given as follows:

---

[9] Stuter, L. (1996). *Delphi Technique, What is it ?* Retrieved March 3, 2007 from http://www.learn-usa.com/transformation_process/acf001.htm

[10] Boehm, B., Abts, C., Chulani,S. (2000). Software Development Cost Estimation Approaches – A Survey. *Annals of Software Engineering*, 10(1-4), 177-205.

[11] Kulk, G.P., and Verhoef, C. (2008). Quantifying Requirements Volatility Effects. *Science of Computer Programming*, 72(3), 136-175.

$$r = \left(\sqrt[t]{\frac{SizeAtEnd}{SizeAtStart}} - 1 \right) \times 100 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

which translates to

$$r = \left(\sqrt[t]{\frac{SLOCAtEnd}{SLOCAtStart}} - 1 \right) \times 100 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

where *t* is the time period in years during which the estimates were observed

However, SLOC is not a suitable proxy for measuring requirements volatility because more often than not, it is dependent on the type of programming language being used and also does not take COTS into consideration, we proposed an alternative proxy such as Function Points, which is a better metric for the size of the software requirements irrespective of how the software will be developed.

$$r = \left(\sqrt[t]{\frac{FuncPointAtEnd}{FuncPointAtStart}} - 1 \right) \times 100 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

4.2 Refining Volatility Estimates

Volatility refinement based on the Dempster-Shafer Theory on Evidence was a key aspect of the framework proposed in (Olagbemiro 2008). Since volatility is a key input parameter needed for Real Options analysis, we attempt to overcome the complexity of volatility estimation by proposing the use of Dempster-Shafer Theory on Evidence, a technique first proposed for application in the domain of *sensor fusion.* It is a mathematical theory of evidence, based on *belief functions* and *plausible reasoning*, which is used to combine separate pieces of information (evidence) to calculate the probability of an event. We posit that it could be used to address both aleotoric and epistemic uncertainties inherent in software-related capital investments by “*fusing*” and reducing uncertainties to the maximum extent possible as they become revealed thereby facilitating a more accurate estimate of the risks propagated by uncertainty and allowing us to develop the appropriate option in response based on a more accurate volatility measure.

We choose to use DST because while Bayesian inference requires all unknowns to be represented by probability distributions, which awkwardly implies the probability of an event for which we are completely ignorant, DST takes over by introducing belief functions to distinguish ignorance and randomness by assigning probability mass to subsets of parameter space, so that randomness is represented by the probability distribution and uncertainty is represented by large subsets (Gelman, 2008)[12]. In other words while Bayesian theory requires probabilities for each uncertainty of interest, the theory of belief functions provides a non-Bayesian way of using mathematical probability to quantify subjective judgments (Shafer, 1996, Chapter 7)[13]. It

---

[12] Gelman, A. (2006). The boxer, the wrestler, and the coin flip: A paradox of robust Bayesian inference and belief functions. *The American Statistician*. 60(2), 146-150.

[13] Shafer, G. (1996) The Art of Causal Conjecture. MIT Press.

measures degrees of belief [or confidence] for one uncertainty on the probabilities for a related uncertainty.

The premise behind DST is it can be interpreted as a generalization of probability theory where probabilities are assigned to sets as opposed to mutually exclusive singletons. In the case that there is sufficient evidence to permit the assignment of probabilities to single events, the Dempster-Shafer model collapses to the traditional probabilistic formulation where evidence is associated with only one possible event (Sentz & Ferson, 2002)[14]. DST relies on three basic functions: the basic probability assignment function, a primitive of evidence theory which does not refer to probability in the classical sense, and two non-additive continuous measures called *Belief* and *Plausibility* which are both used to combine separate pieces of information (evidence) to calculate the probability of an event, while at the same time defining the upper and lower bounds respectively of an interval that contains the precise probability of a set of interest.

Since evidence can be associated with multiple possible events, e.g., sets of events, the evidence in DST can be meaningful at a higher level of abstraction, a key benefit needed at the strategic decision-making level without having to resort to assumptions about the events within the evidential set. Furthermore the DST model can be used to cope with varying levels of precision regarding information with no further assumptions needed to represent the information as demonstrated during a study in addressing uncertainties in systems (Sentz & Ferson, 2002). We posit that the demonstrated approach also allows for the direct representation of uncertainties associated with software-related capital investments since we can characterize vague inputs as sets or intervals with the resulting output also being a set or an interval.

DST is a theory about two things: 1) Degrees of belief and 2) Weights of evidence, with a key benefit of DST being the ability to represent ignorance in the face of uncertainty especially when there is no information so far. In probability theory, uniform distributions are used to represent ignorance, however the problem with this approach is that we represent the space of possibilities affected by the probabilities we get. The theory of belief functions is based on two ideas:

1. The idea of obtaining degrees of belief for one question from subjective probabilities of a related question,

2. Dumpster's rule for combining such degrees of belief when they are based on independent items of evidence. Degrees of belief obtained in this way differ from probabilities in that they may fail to add to 100%.

Both ideas are consistent with the Real Options pre-conditions as the degrees of belief are established on a frame of discernment meant to address uncertainty. DST starts off by assuming a Universe of Discourse Θ, otherwise known as the Frame of Discernment which is a set of mutually exclusive alternatives. Thus a frame of discernment A of a set of mutually exclusive alternatives or possibilities can be represented as

$\Theta = \{A_{1 \ldots\ldots\ldots} A_n\}$ ……………………………………………………………………*(4)*

[14] Sentz, K., and Ferson, S. (2002). *Combination of Evidence in Dempster-Shafer Theory*, Technical Report SAND 2002-0835, Sandia National Laboratories.

where A_1 through A_n represents the set of possibilities or mutually exclusive alternatives.

A key stipulation of DST is that it should be only used to combine belief functions that represent independent items of evidence. The independence required is simply probabilistic independence applied to the questions for which we have probabilities, rather than directly to the question of interest. In other words it means that the sources of information (or at least their current properties as sources of information) are selected independently from well-defined populations.

Combining information or evidence from multiple sources (historical data and Delphi method) in the form of belief assignments serves to aggregate the information with respect to its constituent parts. Dempster proposed a standard combination rule which can be represented as:

$$m_{12}(A) = \sum_{B \cap C = A} \frac{m_1(B)m_2(C)}{1-K} \text{ when } A \neq \emptyset \text{ ....(5)}$$

$$\text{where } K = \sum_{B \cap C = \emptyset} m1(B)m2(C)$$

which is computed by summing the products of the belief probability assignments (bpa's) of all sets where the intersection is null represents basic probability mass associated with conflict, and $m_{12}(A)$ is calculated from the aggregation of two bpa's m_1 and m_2.

Assuming we have two pieces of evidence, based on say historical data and expert judgment (Delphi Method), we now combine the pieces of evidence from both sources using the Dempster's combination rules by computing the orthogonal sum of both pieces of evidence. First we determine all the pairs of sets whose intersection is A for a given set A such that $A_1 \cap A_2 = A$. We then add up the products of the basic probability assignments $m_1(A_1)$ and $m_2(A_2)$, giving us

$$\sum_{A_1 \cap A_2 = A} m_1(A_1)m_2(A_2) \text{ ....(6)}$$

The orthogonal sum of m_1 and m_2 defined by $m = m_1 \oplus m_2$ could then be given as $m(\emptyset) = 0$ and is demonstrated in the matrix below (Table 2), in which we compute the orthogonal sum of three hypothetical risk factors (Risk1, Risk2 and Risk} affecting a software investment program based on two independent expert assessments.

			Independent Expert 1 Subjective estimates on Objective probabilities		
	Risk Factors		{Risk1} m1= 0.80	{Risk1,Risk2} m1= 0.15	{Risk1,Risk2,Risk3} m1 = 0.05
Independent Expert 2 Subjective estimates on 3 Objective probabilities	{Risk1}	m2 =0.70	m1({Risk1})*m2({Risk1})	m1({Risk1,Risk2})*m2({Risk1})	m1({Risk1,Risk2,Risk3})*m2({Risk1})
	{Risk1,Risk 2}	m2 =0.20	m1({Risk1})*m2({Risk1,Risk2})	m1({Risk1,Risk2})*m2({Risk1,Risk2})	m1({Risk1,Risk2,Risk3})*m2({Risk1,Risk2)
	{Risk1,Risk2,Risk3}	m2 =0.10	m1({Risk1})*m2({Risk1,Risk2,Risk3})	m1({Risk1,Risk2})*m2({Risk1,Risk2,Risk3})	m1({Risk1,Risk2,Risk3})*m2({Risk1,Risk2,Risk3})

Table 2. Orthogonal Sum of Basic Probability Assignments.

Based on the sample matrix (Table 2), we can obtain the resulting three evidence functions.

$m_1 \oplus m_2$({Risk1})
= m_1({Risk1})*m_2({Risk1})} + m_1({Risk1,Risk2})*m_2({Risk1})
+ m_1({Risk1,Risk2,Risk3})*m_2({Risk1}) + m_1({Risk1})*m_2({Risk1, Risk2}) + m_1({Risk1})*m_2({Risk1, Risk2, Risk3})

$m_1 \oplus m_2$({Risk1,Risk2})
= m_1({Risk1,Risk2})*m_2({Risk1,Risk2})
+ m_1({Risk1, Risk2})*m_2({Risk1, Risk2, Risk3})
+ m_1({Risk1, Risk2, Risk 3})*m_2({Risk1, Risk2})

$m_1 \oplus m_2$({Risk1,Risk2,Risk3})
= m_1({Risk1,Risk2,Risk3})*m_2({Risk1,Risk2,Risk3})

Using on the information derived from the matrix we can establish joint beliefs. Any variations between inferred probability assignments based on the mass of evidence under this joint belief and our initial volatility estimates based on our modified Caper Jones' equation (Eqn. 3) would reflect inconsistencies. These variations are captured and used to refine the initial probability estimates to reflect the new "findings" which are then modeled using a Monte Carlo simulation to derive new estimates for the requirements volatility and an overall volatility for the software acquisition effort..

5. Applying the Real Options Valuation Framework

In an attempt to validate our proposed approach, we applied the framework to the software component FCSN (Future Combat Systems Network) of U.S. Army Future Combat System (FCS). The decision to select this case study as a validation mechanism was based on the recent nature of the project, the high-risks associated with software development due to the advanced technologies involved, the challenge of networking all of the FCS subsystems together so that FCS-equipped units can function as intended and the associated outcome had a Real Options approach been applied. This section summarizes our study. Readers can refer to (Olagbemiro 2008) for the details of the study.

5.1 Development of a Business Case

We used a traditional discounted cash flow model to obtain a net present value (NPV) in terms of five high-level determinants (Erdogmus & Vandergraaf, 1999)[15]:

$$NPV = \sum \frac{(C_t - M_t)}{(1+r)^t} - I$$

[15] Erdogmus, H. and Vandergraaf, J. (1999). Quantitative Approaches for Assessing the Value of COTS-centric Development. Institute for Information Technology. Proc. Sixth International Symposium on Software Metrics (METRICS'99). Boca Raton, Florida, USA. pp. 279-291

where ***I*** is the (initial) *development cost of the FCSN*
t is the (initial) *development time* or time to deploy the FCSN.
C is the *asset value of the FCSN* over time *t*
M is the *operation cost* of the FCSN over time *t*
r is the rate at which all future cash flows are to be discounted (the *discount rate*).

A NPV of $6.4 trillion[16] was computed for the FCSN using estimated values based on key assumptions in (Olagbemiro, 2008).

5.2 Identification of Uncertainties and Risk Quantification

Using publicly available information (GAO Report 08-467sp, 2008)[17], we determined that requirements uncertainty fostered by technology maturation issues plagued the FCSN program and resulted in the following uncertainties:

1. Requirements uncertainties
2. Integration uncertainties
3. Performance uncertainties
4. Estimation uncertainties (size and cost of the software)
5. Scheduling uncertainties.

Hence, we decided to develop Real Options to mitigate the risk due to requirements change. Due to the lack of publicly available historical data for the FCSN program, data from the Joint Strike Fighter program was fitted and utilized as a source of historical information for comparative purposes. The risk of requirements changes in the FCSN program was estimated to be 12% (as oppose to 0.28% for the JSF program which is one fifth the size of the FCSN program) using (Eqn. 1)[18].

We used requirements volatility to quantify the effect of the risk as variations in the returns associated with the investment. We ran Monte Carlo simulation of the risk model using the Risk Simulator software, taking into account interdependencies between the risk variables to emulate all potential combinations and permutations of outcomes. The analysis indicated that requirements volatility introduced an overall volatility of 0.0866% in the FCSN program. The volatility of 0.0866% resulted in a reduction in the NPV of the FCSN program from $6.4 trillion to $6.1 trillion. This reduction in NPV is as a result of the potential of increased costs in light of the

---

[16] NPV of $6.4 trillion is computed based on (1) Value of the FCSN program, (future value less operating costs, i.e. sum of (C – M), = $10 trillion), (2) Initial development cost I = $163.7 billion, (3) r = 3%, and (4) Time t to develop the FCSN = 13 years.

[17] U.S. General Accounting Office. (2008). Defense Acquisitions, Assessments Selected Weapon Programs. (U.S. GAO Report GAO 08-467sp). Washington, DC. U.S. Government Printing Office.

[18] The requirements volatility of 12% was computed based on start and ending SLOC for the FCSN program. SLOC is used for demonstration purposes only. A more suitable metric such as function points is recommended.

risks facing the FCSN program, which ultimately reduces the value of the investment effort from a financial point of view.

To improve the accuracy of the volatility estimates, we chose to refine the volatility using the DST. This is accomplished by establishing “belief functions” that reflect the “degrees of belief” between our NPV estimates in light of the risks posed by requirements uncertainty and the FCSN cost estimates provided by two independent sources, the Cost Analysis Improvement Group (CAIG) and the Institute of Defense Analysis (IDA). The independent belief functions based on the CAIG and IDA which inferred basic probability assignments associated with each of the FCSN risk factors (requirements, integration, estimation risk etc…) were combined using an orthogonal matrix to determine the most probable beliefs for the set of risk factors. Where the combined functions reflected “belief” in our estimates, our estimates were considered to be valid and were left untouched, and in situations where the combined belief functions reflected conflict with our estimates, our estimates were revised accordingly. We ran the Monte Carlo simulation of the model with the revised risk estimates again. Based on the risk of requirements uncertainty presented in the FCSN, a resulting “refined” volatility of 0.0947% was obtained. The derived volatility which reflects an increase from the initial volatility estimate of 0.0866% results in a further reduction of NPV of the FCSN program from $6.1 trillion to $5.7 trillion. Details of the computation can be found in (Olagbemiro 2008).

5.3 Options Development

Given that the FCS software effort has been decomposed into the following six components: Combat Identification, Battle Command and Mission Execution, Network Management System, Small Unmanned Ground Vehicle, Training Common Component, and Systems of Systems Common Operating Environment, we consider a hypothetical scenario in which we assume that of the six component systems, the Systems of Systems Common Operating Environment, is not facing uncertainty while the other five software components are facing uncertainty. We proceeded and developed two different options to address this scenario: (1) Compound Option and (2) Deferment Option.

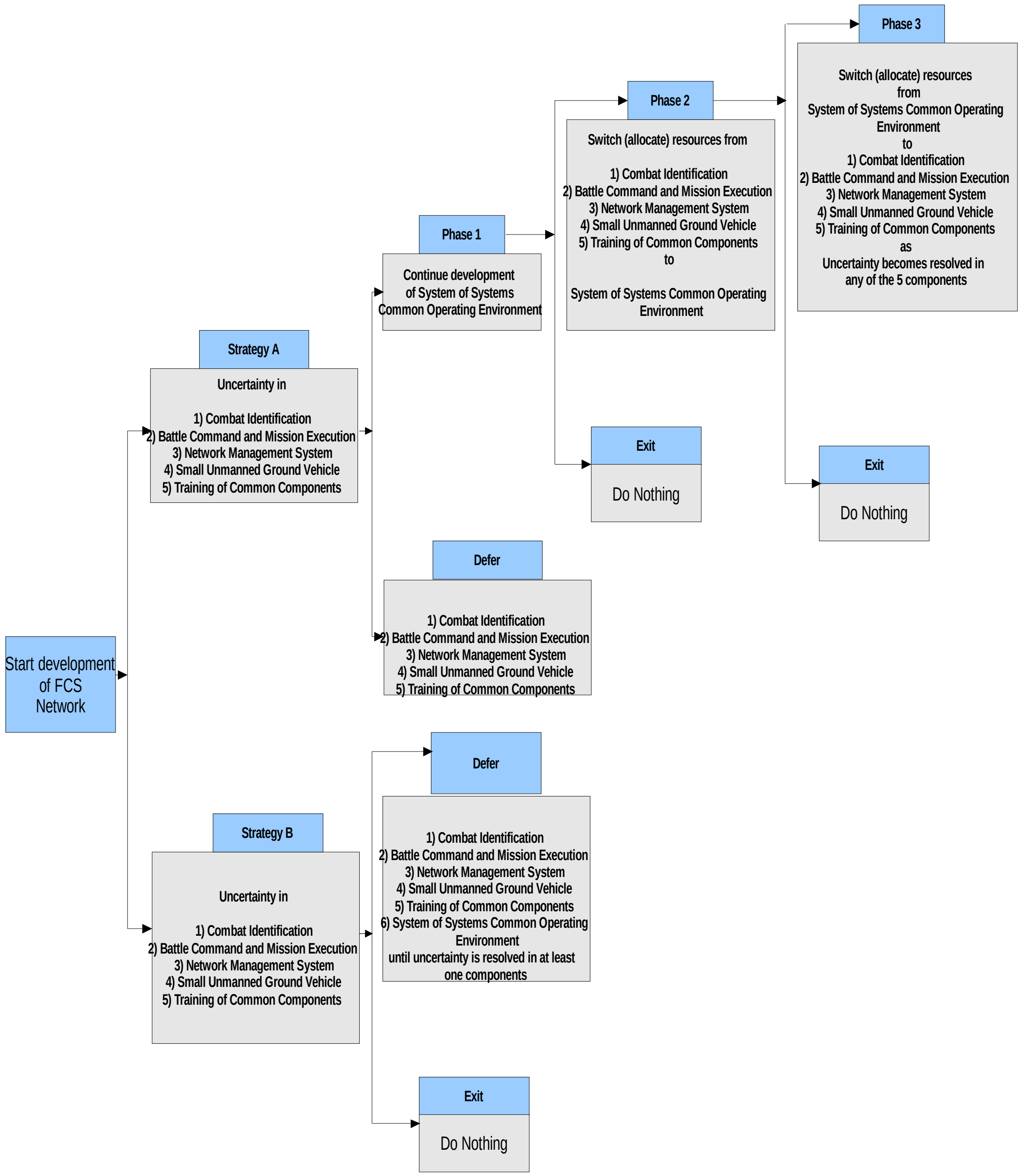


Figure 5. FCS Strategy Tree depicting Strategy A and B for given Scenario

(1) Strategy A - the Compound Option

In the event that at least one of the software components is not facing requirements uncertainty, with all the others facing requirements uncertainty, an option could be developed to scale down the resources/staff allocated to the software components facing requirements uncertainty. The staff could then be switched to work on the software component that is not facing requirements uncertainty, while the uncertainties in the other components are addressed using our uncertainty elicitation model[19]. We therefore frame the real options in this case as: an *Option to Contract and Scale Down* from an uncertain system, *Option to Switch* resources to another system, *Options to Expand and Scale Up* staff assigned to the development of a system not facing uncertainty (shown as Strategy A in Figure 5). This is essentially a compound option, an option whose "exercise" is contingent on the execution of the preceding option.

(2) Strategy B - the Deferment Option

In the event that five out of the six software components are facing requirements uncertainty, then an option could be developed to stop and defer all development to include the development of the software component that is not facing requirements uncertainty for a specified period until uncertainty is resolved (shown as Strategy B in Figure 5). This is an *Option to Wait and Defer*.

5.4. Options Valuation

We utilize the Real Options Super Lattice Solver (SLS) 3.0 software developed by Real Options Valuation, Inc. for the task.

(1) Strategy A

The Real Options SLS software was populated based on the following underlying values: (1) Development/Implementation cost of FCSN is $163.7 billion, (2) Value of underlying asset is $6.4 trillion, (3) The risk-free rate is 3.0%, (4) Volatility of the project is 0.0947, (5) Duration of software development is 13 years, and (6) Lattice steps was set to 300. The value of the underlying asset was computed as $6.4 trillion and the option analysis of the value of the option under Strategy A returned a value of $6.27 trillion.

(2) Strategy B

In Strategy B, which calls for a "defer and wait approach", an assumption is made that the duration for deferment option would be 3 years. We set up our model using the same assumptions used in strategy A, but set the duration of the Deferment Option to 3 years. The value of the underlying asset was computed as $6.4 trillion ands the option analysis returned a value of $6.25 trillion.

[19] Note: The assumption with this approach is that the software component development effort which the staff engineers are being reallocated to work on is not already behind schedule and hence does not violate Brooks Law.

5.5. Investment Valuation

Given the option value of $6.27 trillion under Strategy A, the intrinsic value of the compound option is determined to be $6.27 trillion – $5.7 trillion = $570 billion. Under Strategy B, the intrinsic value of the deferment option is determined to be $6.25 trillion – $5.7 trillion = $550 billion. This implies is that under both Strategies A and B, the software executive should be willing to pay no more than (and hopefully *much less than*) the option value of $570 billion and $550 billion respectively in addition to the initial investment cost of $163.7 billion to increase the chances of receiving the projected NPV of $6.27 trillion under Strategy A and $6.25 trillion under Strategy B for the FCSN as opposed to the current projected $5.7 trillion in light of the risks caused by the uncertainties in five of the six software components. This premium would also include the administrative costs associated with exercising an option from an integrated logistics support point of view, i.e. costs associated with contractual agreements, software development retooling costs, costs associated with infrastructure setup of the infrastructure etc.

In analyzing both strategies, Strategy A is more attractive than Strategy B. Instead of waiting for another 3 years at an additional cost of up to $550 billion (after which uncertainty would hopefully have been resolved) and then proceeding to spend $163.7 billion at once to develop all six software components, the staged phase approach in strategy A calls for spending up to $570 billion for the option up front plus some of the $163.7 billion for the Systems of Systems Common Operating Environment component, and then investing more over time as the requirements are firmed up for the other five components. Therefore under these conditions, Strategy A which employs the compound sequential options is the optimal approach.

6. Conclusion

Uncertainties associated with software-related capital investments lead to unnecessary and sometimes preventable risks. As DoD often sets optimistic requirements for weapons programs that require new and unproven technologies, the application of the real options valuation methodology would be beneficial as it would enable the DoD to incorporate the appropriate *strategic options* into the acquisition contracts. The options would serve as a contract between the software executive and the contractor—in the case of a government acquisition—to buy or sell a specific capability known as the options on the underlying project. The proposed real options valuation approach is able to overcome the limitations of traditional valuation techniques by utilizing the best features of traditional approaches and extending their capabilities under the auspices of managerial flexibility. The explicit uncertainty elicitation task, the development of options to hedge against the risk, and the timely execution of the options as they appear will allow decision makers to better balance customer requirements as dictated by operational needs within financial viability and schedule constraints and manage risks proactively.

THIS PAGE INTENTIONALLY LEFT BLANK

2003 - 2009 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning
- Capital Budgeting for DoD

- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-tem Attrition
- Retention
- The Navy's Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness
- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC Aegis Microwave Power Tubes

- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org

Application of Real Options Theory to Software-Intensive System Acquisition

Man-Tak Shing, Albert Olagbemiro, Johnathan Mun

Naval Postgraduate School

Acknowledgement and Disclaimer

- The research was funded in part by a grant from the NPS Acquisition Research Program - OUSD_08.
- The views and conclusions in this talk are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

Outline

- Introduction
  - Uncertainties in Software-Intensive Weapon Acquisition
- Real Options Theory Approach to Risk Management
  - Real Options Framework
- Addressing Uncertainties
  - Uncertainties Elicitation and Quantification
  - Refining uncertainties quantification with Dempster Shaffer' Theory
- Options Development and Valuation
- Discussion

Uncertainties in Software-Intensive Weapon Acquisition

- Uncertainties in the <u>system/software requirements</u> and <u>new/unproven technologies</u> have been a leading problem in almost all software-intensive weapon systems acquisition
- We need a *proactive* way to manage the risks caused by these uncertainties
  - *Managing risk at the acquisition level, instead of at the development level*
  - *During the investment decision-making activities prior to the commitment to acquire and/or develop a software system*

Real Options Theory Approach to Risk Management

- Value strategic software acquisitions decisions as a portfolio of options or choices in real "assets"
- Provide the software manager with time-deferred and flexible choices (options) regarding future risks or changes of software

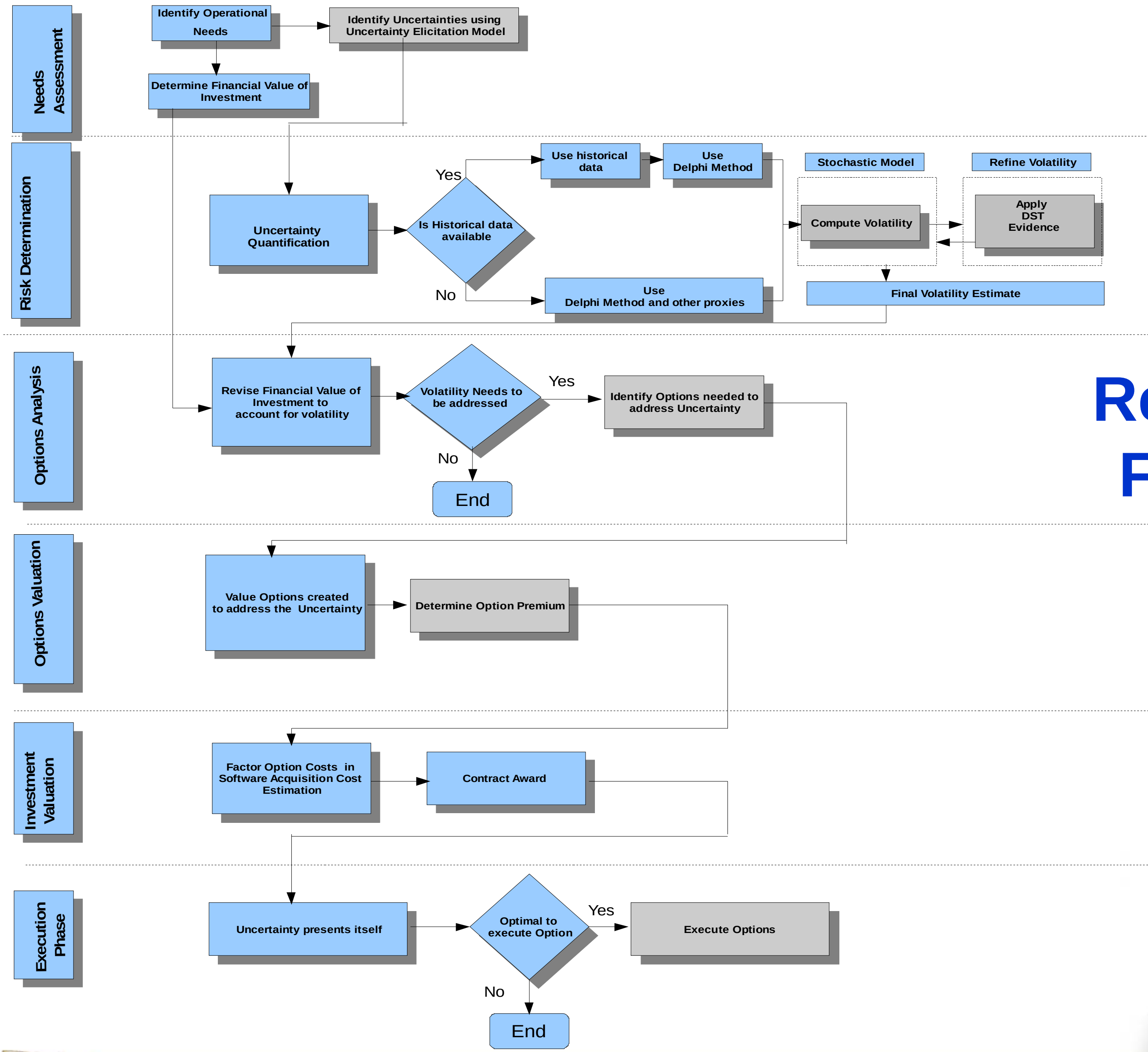
Pre-conditions for Effective Use of Real Options

- The existence of a basic financial model for “Real” assets.
- The existence of uncertainties.
- The uncertainties must introduce risks which directly impact the project.
- Management must have the flexibility or option to make *mid course* corrections when actively managing the project.
- Management must have the wisdom to execute the Real Options when it becomes optimal to do so.

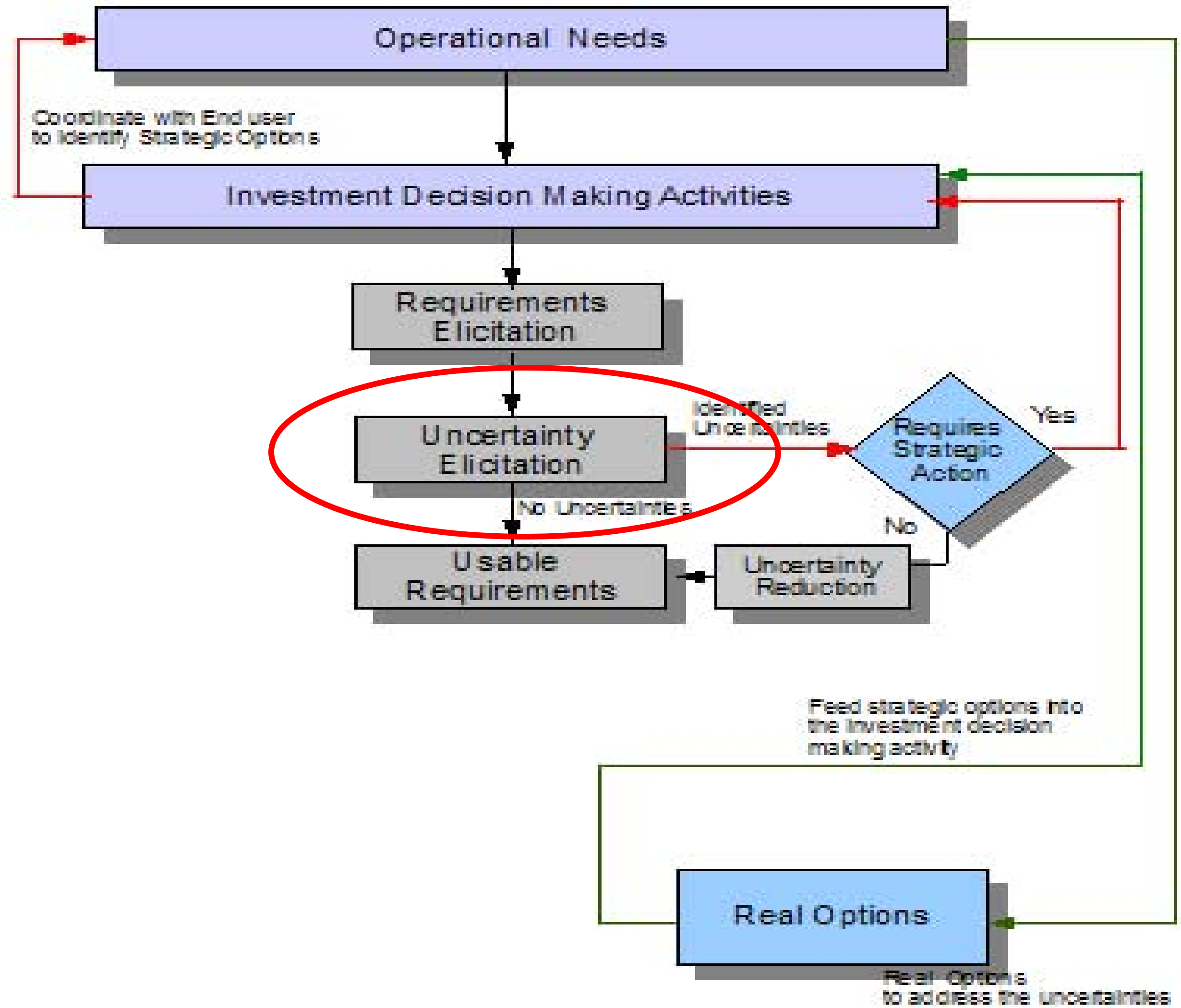
Meeting The Pre-conditions

- Establish compliance with Real Options methodology pre-conditions 1 - 3.
  - Develop a financial model for the benefits/cost of software acquisition
  - Identify the uncertainties in software acquisition
  - Quantify associated risks
- Option Identification.
- Option Valuation.

Real Options Framework



Identifying Uncertainties



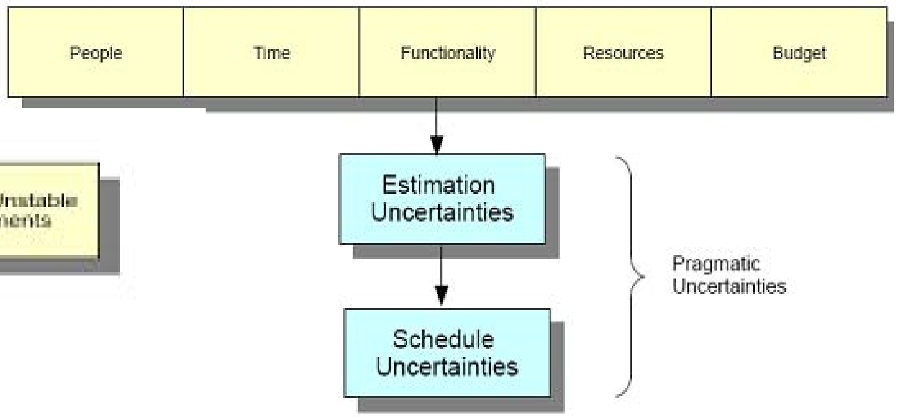
Managerial and Technical Uncertainties

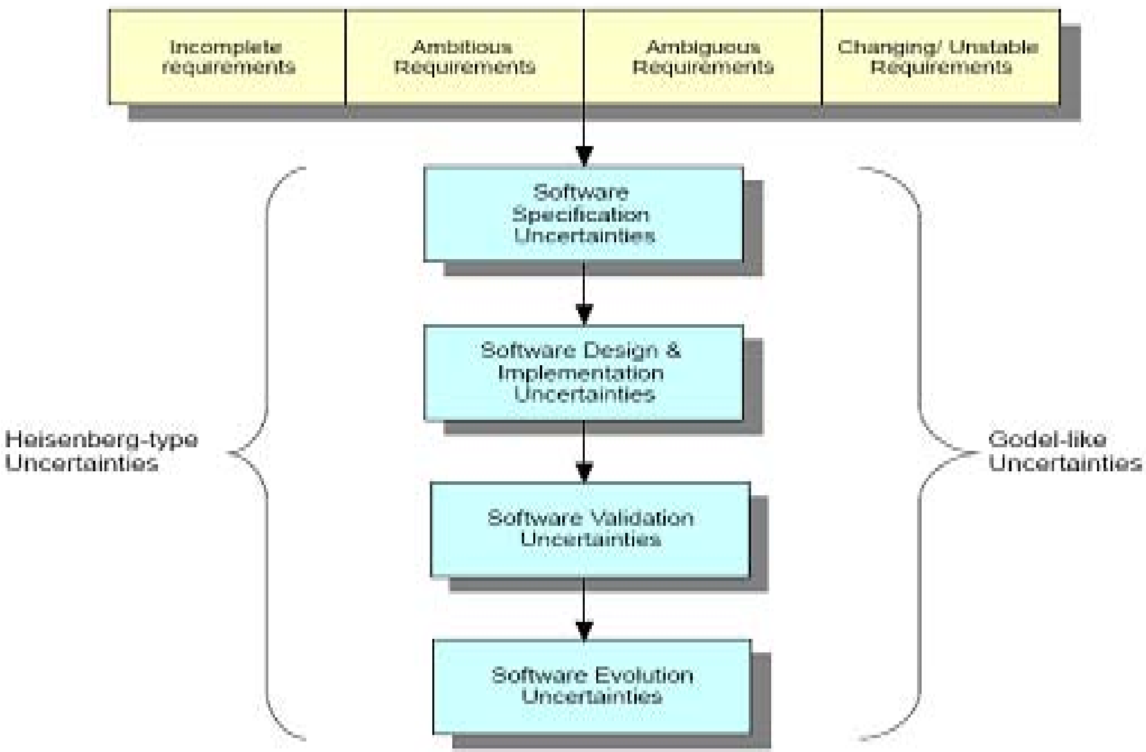
Technical Uncertainties:

Requirements uncertainties

Integration uncertainties

Performance uncertainties





Managerial Uncertainties:

Estimation uncertainties (size/cost)

Scheduling uncertainties

Quantifying Uncertainty

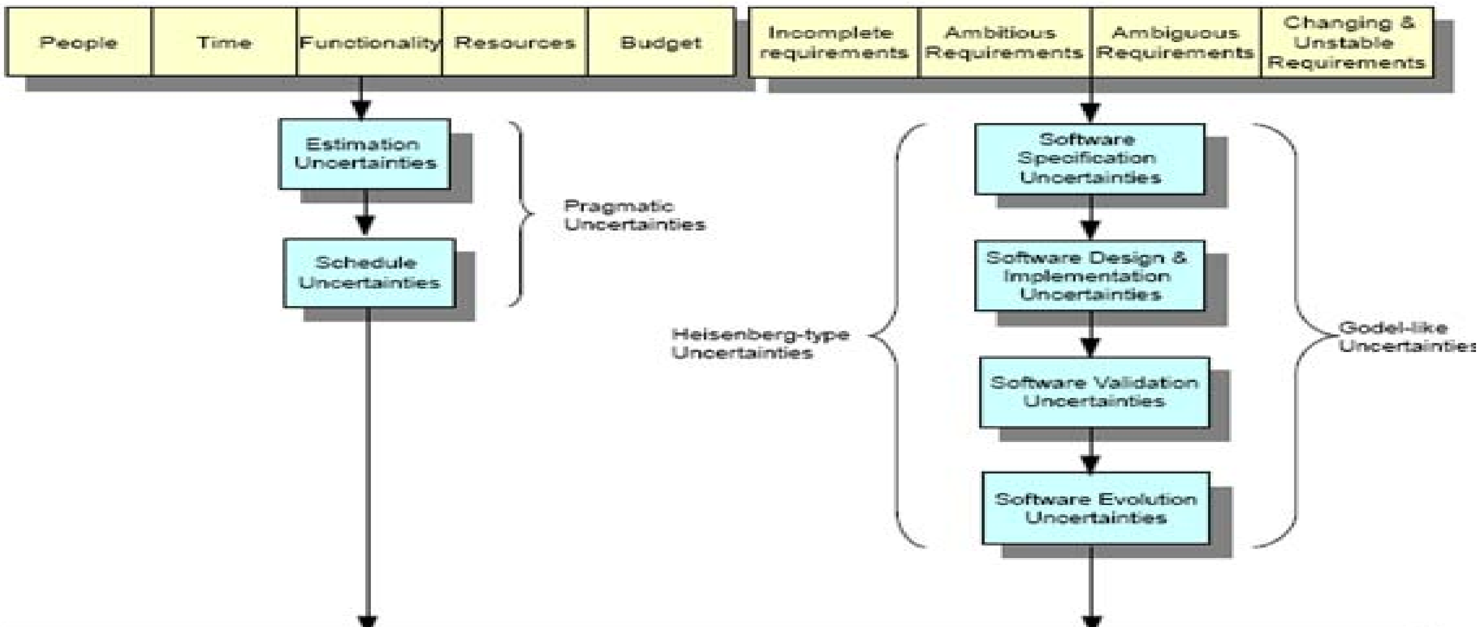
- Equate the uncertainties of the current investment effort to a quantifiable property (risk factor) in order to gauge the magnitude/impact of the risk on the underlying asset
  - E.g., determine the rate of requirements change and then estimate the effect of the requirements change on the program

Determine The Rate of Requirements Change

- Based on historical evidence from existing project or similar projects
- Based on expert opinions obtained using the Delphi method in the absence of historical data
- For example, to determine the rate of the requirements change in the Future Combat System Network (FCSN):
  - We use the Joint Strike Fighter (JSF) program as a proxy.
  - We apply the data from JSF to Capers Jones' formula and obtain an estimate of 12%.

Estimation of Requirements Volatility (RV)

- We use RV to quantify the effect of the risk as variations in the returns associated with the investment
- Determine RV by running a Monte Carlo simulation of the risk model using the Risk Simulator software



				Risk Factors							
NPV of Future Returns on a Investment of $163.7 billion in an Asset Valued at $10 Trillion	Current Requirements (SLOC)	Planned Schedule (Months)	Planned Costs	Requiements Creep Risk	Intergration Risk	Schedule Overrun	Performance Risk	Software Cost Estimation	Unit Cost	Monthly Cost	Final Costs Due to Risk Factors
$6,138,586,402,574.14	95000000	156	$163,700,000,000.00	190.90%	0.56%	-0.02%	9.38%	91.37%	$1,723.16	$1,364,166,666.67	$670,926,997,357.64

Reduction in FCS Returns from $6.4 Trillion to $6.1 Trillion (New Value)

Volatility = 0.0866 %

Applying Dempster-Shaffer Theory (DTS) to Refine RV Estimates (1/3)

- Also known as the Theory of Belief Functions
  - A generalization of the Bayesian theory of subjective probability
- DTS is based on two ideas
  - The idea of obtaining degrees of belief for one question from the subjective probabilities for a related question
  - Dempster' rule for combining such degrees of belief when they are based on independent items of evidences

Applying DTS to Refine RV Estimates (2/3)

- DST is well suited to address software uncertainties
  - Reduce epistemic uncertainties by increasing one's knowledge of the problem at hand
    - Epistemic uncertainties are reducible uncertainties due to lack of knowledge, lack of information and our own and others' subjectivity concerning an issue
  - Ability to represent ignorance in the face of uncertainty when there is no information

Applying DTS to Refine RV Estimates (3/3)

- For example, to improve the RV estimates of the Future Combat System Network (FCSN):
  - We use data from two independent source, Cost Analysis Improvement Group (CAIG) and Institute for Defense Analysis (IDA) to derive degrees of belief of the FCSN risk factors, then we use Dempster's rule to combine the degrees of beliefs
  - Where the combined functions reflected "belief" in our estimates, our estimates were considered to be valid and were left untouched, and in situations where the combined belief functions reflected conflict with our estimates, our estimates were revised accordingly.

Reduction in FCS Returns from \$6.4 Trillion to \$5.7 Trillion (Refined Value)

Volatility = 0.0947 %

Identifying Options (1/2)

- Scenario
  - FCSN can be broken down into 6 component systems
  - We consider a hypothetical scenario in which we assume that of the six component systems, the Systems of Systems Common Operating Environment, is not facing uncertainty while the other five software components are facing uncertainty

Identifying Options (2/2)

- We proceeded and developed two different options to address this scenario:
  - Strategy A - Compound Option
    - *Option to Contract and Scale Down* from an uncertain system,
    - *Option to Switch* resources to another system
    - *Options to Expand and Scale Up* staff assigned to the development of a system not facing uncertainty
  - Strategy B - Deferment Option
    - *Option to Wait and Defer, including the one not facing uncertainty*

Options Valuation using the Real Option SLS software (1/2)

- Strategy A - Compound Option
  - Input parameters:
    (1) Development/Implementation cost of FCSN is $163.7 billion, (2) Value of underlying asset is $6.4 Trillion, (3) The risk-free rate is 3.0%, (4) Volatility of the project is 0.0947, (5) Duration of software development is 13 years, and (6) Lattice steps was set to 300.
  - The value of the underlying asset was computed as $6.4 Trillion and the option analysis of the value of the compound option returned a value of $6.27 Trillion.
  - The intrinsic value of the compound option = $6.27 Trillion – $5.7 Trillion = $570 Billion.

Options Valuation using the Real Option SLS software (2/2)

- Strategy B - Deferment Option
  - Input parameters:
    (1) Development/Implementation cost of FCSN is $163.7 billion, (2) Value of underlying asset is $6.4 Trillion, (3) The risk-free rate is 3.0%, (4) Volatility of the project is 0.0947, (5) Duration for deferment option would be 3 years, and (6) Lattice steps was set to 300.
  - The value of the underlying asset was computed as $6.4 Trillion and the option analysis of the value of the deferment option returned a value of $6.25 Trillion.
  - The intrinsic value of the deferment option = $6.25 Trillion – $5.7 Trillion = $550 Billion.

Outcome of Real Option Analysis

- Current Approach
  - Net Present Value of FCS Network = $6.4 Trillion
  - Investment Cost = $163.7 Billion
  - Considers effect of risk, Net Present Value of FCS Network = $5.7 Trillion
- Option Premium for Strategy A = $570 Billion
- What this Means
  - Pay an extra premium that is less than $570 Billion today
  - Reduce the chance of losing
    $6.4 Trillion - $5.7 Trillion = $700 Billion

Conclusion

- We present a proactive approach to address the risks associated with software-related capital investments that emphasizes planning for and paying for risk up front
- Need to validate the proposed approach with detailed data from real acquisition program
- Need to create a repository of historical data to serve as a basis of comparison with current/future acquisition programs
- Formalize and create an automated software acquisition decision-making tool explicitly aimed at managing the risks associated with software-related capital investments using our Real Options approach

Backup Slides

Caper Jones' Formula for Rate of Requirements Change

$$r = \left(\sqrt[t]{\frac{SizeAtEnd}{SizeAtStart}} - 1 \right) \times 100$$

Delphi Method

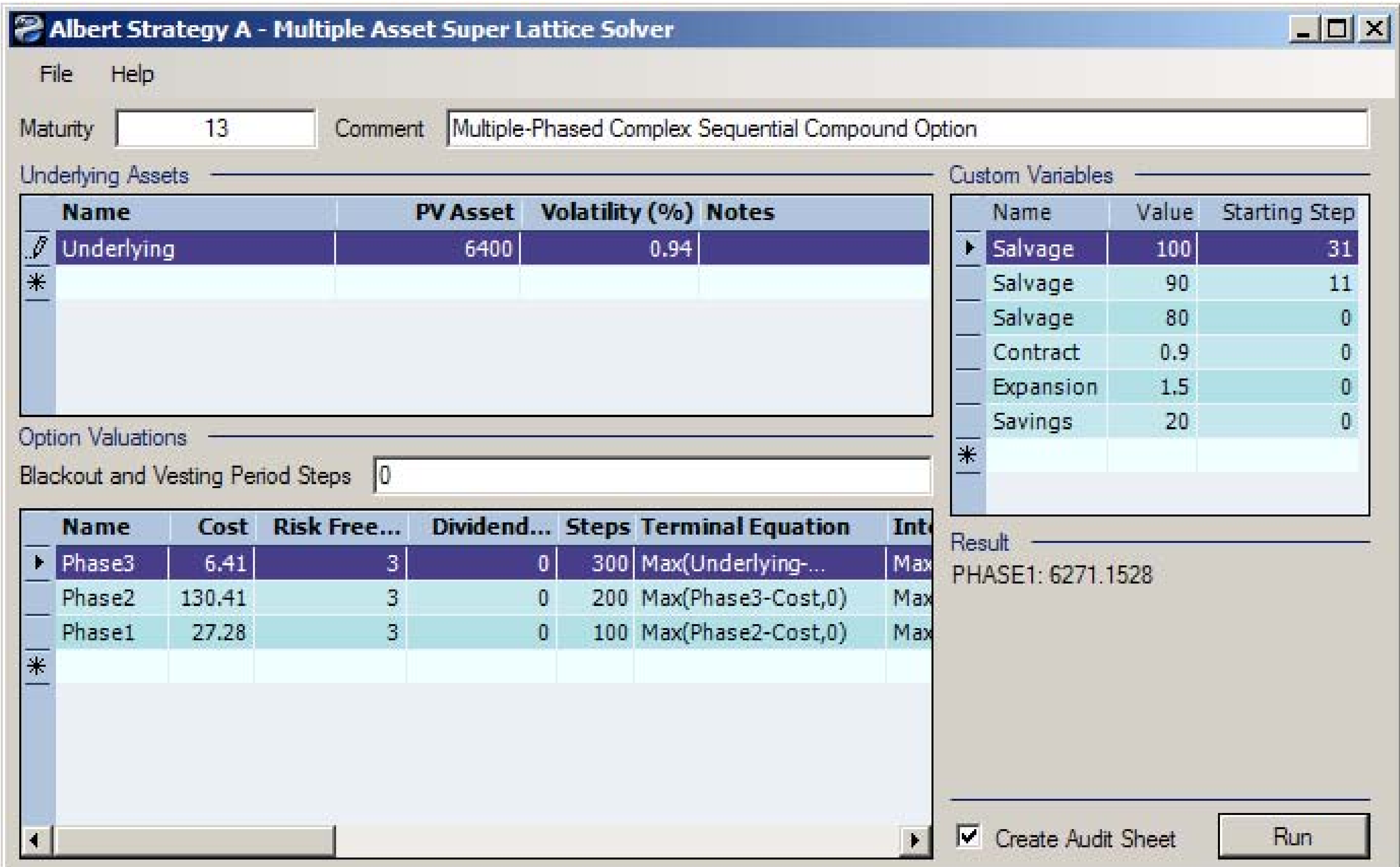
- A methodology for the elicitation of the opinion of an expert or groups of experts to guide decision making by the making predictions about future events
- A three step process of thesis, antithesis, and synthesis
  - In the thesis and antithesis steps, the team of experts present their opinion or views on the given subject, establishing views and opposing views.
  - Consensus is ultimately reached during the synthesis phase as opposing views are brought together to form the new thesis.

Applying DTS to Refine RV Estimates

- We ran the Monte Carlo simulation of the model with the revised risk estimates again.
- Based on the risk of requirements uncertainty presented in the FCSN, a resulting “refined” volatility of 0.0947% was obtained.
- The derived volatility which reflects an increase from the initial volatility estimate of 0.0866% results in a further reduction of NPV of the FCSN program from $6.1 Trillion to $5.7 Trillion.

Real Options Input Parameters

Symbol	Real Option on Software Acquisitions Project	Description
S	Value of underlying Asset: (Asset Price)	Current Value of expected cash flows. (Expected benefits realized from investing in the software effort (NPV))
K	Exercise Price / Strike Price	Price at which the created option would be realized (Investment Cost, of investing in options, which is an estimation of the likely costs of accommodating changes)
T	Time-to-expiration	The useful life of the option (Time until the opportunity disappears/ maturity date of the option contract)
r	Risk-free Interest Rate	Risk free interest rate relative to budget and schedule (Interest rate on US Treasury bonds)
cv	Volatility	Uncertainty of the project value and fluctuations in the value of the requirements over a specified period of time (Volatility in requirements, cost estimation and schedule estimation based on DST of Evidence)



Compound Option Model in the Real Options SLS software

FCSN Underlying Asset Value Lattice

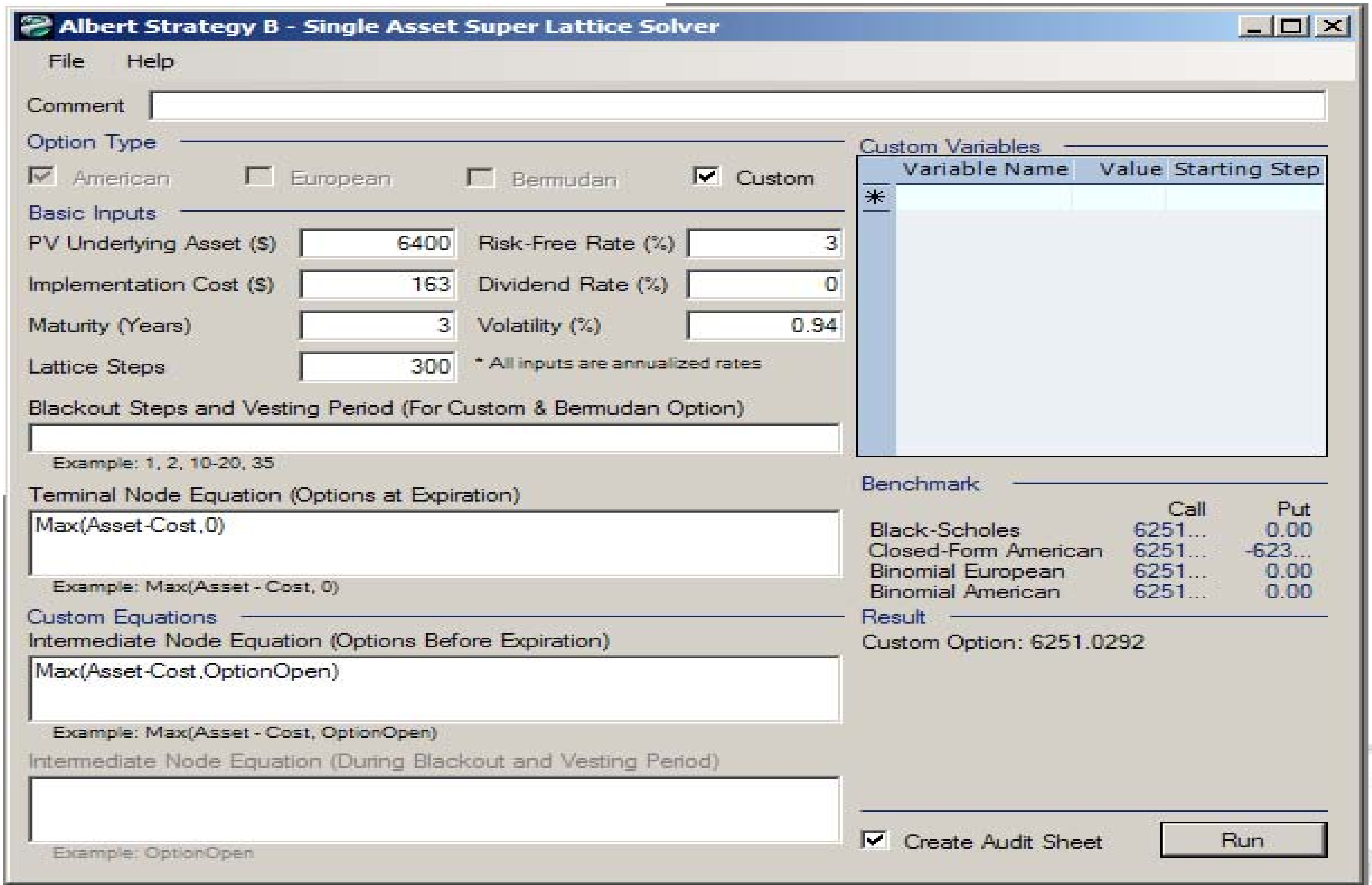
										6526.47
									6513.71	
								6500.97		6500.97
							6488.27		6488.27	
						6475.58		6475.58		6475.58
					6462.92		6462.92		6462.92	
				6450.29		6450.29		6450.29		6450.29
			6437.68		6437.68		6437.68		6437.68	
		6425.10		6425.10		6425.10		6425.10		6425.10
	6412.54		6412.54		6412.54		6412.54		6412.54	
6400.00		6400.00		6400.00		6400.00		6400.00		6400.00
	6387.49		6387.49		6387.49		6387.49		6387.49	
		6375.00		6375.00		6375.00		6375.00		6375.00
			6362.54		6362.54		6362.54		6362.54	
				6350.10		6350.10		6350.10		6350.10
					6337.69		6337.69		6337.69	
						6325.30		6325.30		6325.30
							6312.93		6312.93	
								6300.59		6300.59
									6288.28	
										6275.98

Lattice of Underlying Asset (FCS Network)

Strategy A Option Valuation Lattice

										6395.93
									6383.34	
								6370.78		6370.44
							6358.24		6357.90	
						6345.73		6345.39		6345.05
					6333.24		6332.90		6332.56	
				6320.77		6320.43		6320.10		6319.76
			6308.33		6307.99		6307.66		6307.32	
		6295.91		6295.58		6295.24		6294.90		6294.56
	6283.52		6283.18		6282.85		6282.51		6282.17	
6271.15		6270.82		6270.48		6270.14		6269.81		6269.47
	6258.47		6258.14		6257.80		6257.46		6257.13	
		6245.82		6245.48		6245.15		6244.81		6244.47
			6233.19		6232.85		6232.52		6232.18	
				6220.58		6220.25		6219.91		6219.57
					6208.00		6207.66		6207.33	
						6195.44		6195.11		6194.77
							6182.91		6182.57	
								6170.40		6170.06
									6157.91	
										6145.45

Option Valuation Lattice of Strategy A



Deferment Option Model in the Real Options SLS software

Strategy B Option Valuation Lattice

										6311.03
									6305.00	
								6298.98		6298.89
							6292.97		6292.88	
						6286.96		6286.87		6286.78
					6280.96		6280.87		6280.78	
				6274.96		6274.87		6274.78		6274.69
			6268.97		6268.88		6268.79		6268.70	
		6262.98		6262.89		6262.80		6262.71		6262.62
	6257.00		6256.91		6256.82		6256.73		6256.65	
6251.03		6250.94		6250.85		6250.76		6250.67		6250.58
	6244.97		6244.88		6244.79		6244.70		6244.61	
		6238.92		6238.83		6238.74		6238.65		6238.56
			6232.87		6232.78		6232.69		6232.60	
				6226.83		6226.74		6226.65		6226.56
					6220.80		6220.71		6220.62	
						6214.77		6214.68		6214.59
							6208.74		6208.65	
								6202.72		6202.63
									6196.71	
										6190.70

Option Valuation Lattice of Strategy B